IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION

STRAGENT LLC, et al.,	§	
	§	**NO. 6:11cv421 LED-JDL**
v.	§	
	§	**PATENT CASE**
INTEL CORPORATION.	§	

MEMORANDUM OPINION AND ORDER

This claim construction opinion construes the disputed claim terms in U.S. Patent Nos. 6,848,072 ("the '072 patent"); 7,028,244 ("the '244 patent"); and 7,320,102 ("the '102 patent"). Plaintiffs Stragent, LLC and TAG Foundation ("collectively, Stragent") allege Defendant Intel Corporation ("Intel") infringes the patents-in-suit. The parties have presented their claim construction positions (Doc. Nos. 79, 82 & 88). On March 7, 2013, the Court held a claim construction hearing. For the reasons stated herein, the Court adopts the constructions set forth below.

OVERVIEW OF THE PATENTS

The patents-in-suit are related. The '244 patent is a continuation of the '072 patent, and the '102 patent is a continuation of the '244 patent. PLTFF'S BRIEF AT 4 n.9. All the patents-in-suit share a common title, inventor, and figures. *Id.* Aside from a few typographical differences, the patents share the same specification. *Id.* at 4-5.

The invention disclosed in the patents-in-suit relates "to network devices designed to implement network protocols." '072 patent at 1:15-16; '244 patent at 1:19-20; '102 patent at 20-21.[1] Particularly, the invention is directed to the use of "hardware implemented [Cyclic

[1] Because the patents-in-suit, to a large extent, share a specification, the Court cites the '072 patent for ease of reference.

Redundancy Check (CRC)] instructions in a network processor" to detect error in transmitted packets of data. *See* '072 patent at 1:45-65. The purpose of the invention is to "perform CRC calculations in an efficient manner." *Id.* at 1:57-58.

Stragent asserts Intel infringes Claims 1-10, 12-16, and 19-22 of the '072 patent; Claims 1-8 and 10-11 of the '244 patents; and Claims 1-3 and 5-9 of the '102 patent.[2] PLTFF'S BRIEF AT 9. Claim 6 of the '072 patent is set forth below as a representative claim with disputed claim terms set forth in bold:

> A network device comprising:
>
> an **instruction store** including at least one Cyclic Redundancy Check (CRC) instruction relating to a CRC operation; and
>
> an arithmetic logic unit (ALU) connected to the instruction store, the ALU including at least one **CRC circuit** for generating a **CRC result** based on hardwired CRC polynomials, the ALU receiving input data for the **CRC operation** and the CRC instruction, and in response to the **CRC instruction**, generating the CRC result using the CRC circuit, the input data, and a selected one of the hardwired polynomials, the selected hardwired polynomial being selected based on the CRC instruction.

'072 patent at 6:31-45 (Claim 6).

CLAIM CONSTRUCTION PRINCIPLES

"It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (quoting *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). The Court examines a patent's intrinsic evidence to define the patented invention's scope. *Id.* at 1313-1314; *Bell Atl. Network Servs., Inc. v. Covad*

[2] The Court provisionally construed many of the terms in dispute in *Stragent, LLC v. Freescale Semiconductor Inc.*, No. 6:10cv224 (Doc. No. 141) (Sept. 23, 2011).

Commc'ns Group, Inc., 262 F.3d 1258, 1267 (Fed. Cir. 2001). Intrinsic evidence includes the claims, the rest of the specification and the prosecution history. *Phillips*, 415 F.3d at 1312-13; *Bell Atl. Network Servs.*, 262 F.3d at 1267. The Court gives claim terms their ordinary and customary meaning as understood by one of ordinary skill in the art at the time of the invention. *Phillips*, 415 F.3d at 1312-13; *Alloc, Inc. v. Int'l Trade Comm'n*, 342 F.3d 1361, 1368 (Fed. Cir. 2003).

Claim language guides the Court's construction of claim terms. *Phillips*, 415 F.3d at 1314. "[T]he context in which a term is used in the asserted claim can be highly instructive." *Id.* Other claims, asserted and unasserted, can provide additional instruction because "terms are normally used consistently throughout the patent." *Id.* Differences among claims, such as additional limitations in dependent claims, can provide further guidance. *Id.*

"[C]laims 'must be read in view of the specification, of which they are a part.'" *Id.* (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995)). "[T]he specification 'is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.'" *Id.* (quoting *Vitronics Corp.v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); *Teleflex. Inc. v.Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002). In the specification, a patentee may define his own terms, give a claim term a different meaning that it would otherwise possess, or disclaim or disavow some claim scope. *Phillips*, 415 F.3d at 1316. Although the Court generally presumes terms possess their ordinary meaning, this presumption can be overcome by statements of clear disclaimer. *See SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc.*, 242 F.3d 1337, 1343-44 (Fed. Cir. 2001). This presumption does not arise when the patentee acts as his own lexicographer. *See Irdeto Access, Inc. v. EchoStar Satellite Corp.*, 383 F.3d 1295, 1301 (Fed. Cir. 2004).

The specification may also resolve ambiguous claim terms "where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone." *Teleflex, Inc.*, 299 F.3d at 1325. For example, "[a] claim interpretation that excludes a preferred embodiment from the scope of the claim 'is rarely, if ever, correct." *Globetrotter Software, Inc. v. Elam Computer Group Inc.*, 362 F.3d 1367, 1381 (Fed. Cir. 2004) (quoting *Vitronics Corp.*, 90 F.3d at 1583). But, "[a]lthough the specification may aid the court in interpreting the meaning of disputed language in the claims, particular embodiments and examples appearing in the specification will not generally be read into the claims." *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571 (Fed. Cir. 1988); *see also Phillips*, 415 F.3d at 1323.

The prosecution history is another tool to supply the proper context for claim construction because a patentee may define a term during prosecution of the patent. *Home Diagnostics Inc. v. LifeScan, Inc.*, 381 F.3d 1352, 1356 (Fed. Cir. 2004) ("As in the case of the specification, a patent applicant may define a term in prosecuting a patent."). The well established doctrine of prosecution disclaimer "preclud[es] patentees from recapturing through claim interpretation specific meanings disclaimed during prosecution." *Omega Eng'g Inc. v. Raytek Corp.*, 334 F.3d 1314, 1323 (Fed. Cir. 2003). The prosecution history must show that the patentee clearly and unambiguously disclaimed or disavowed the proposed interpretation during prosecution to obtain claim allowance. *Middleton Inc. v. 3M Co.*, 311 F.3d 1384, 1388 (Fed. Cir. 2002); *see also Springs Window Fashions LP v. Novo Indus., L.P.*, 323 F.3d 989, 994 (Fed. Cir. 2003) ("The disclaimer . . . must be effected with 'reasonable clarity and deliberateness.'")

(citations omitted)). "Indeed, by distinguishing the claimed invention over the prior art, an applicant is indicating what the claims do not cover." *Spectrum Int'l v. Sterilite Corp.*, 164 F.3d 1372, 1378-79 (Fed. Cir. 1988) (quotation omitted). "As a basic principle of claim interpretation, prosecution disclaimer promotes the public notice function of the intrinsic evidence and protects the public's reliance on definitive statements made during prosecution." *Omega Eng'g, Inc.*, 334 F.3d at 1324.

Although, "less significant than the intrinsic record in determining the legally operative meaning of claim language," the Court may rely on extrinsic evidence to "shed useful light on the relevant art." *Phillips*, 415 F.3d at 1317 (quotation omitted). Technical dictionaries and treatises may help the Court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but such sources may also provide overly broad definitions or may not be indicative of how terms are used in the patent. *Id.* at 1318. Similarly, expert testimony may aid the Court in determining the particular meaning of a term in the pertinent field, but "conclusory, unsupported assertions by experts as to the definition of a claim term are not useful." *Id.* Generally, extrinsic evidence is "less reliable than the patent and its prosecution history in determining how to read claim terms." *Id.*

DISCUSSION

I. "instruction"[3]

Plaintiff's Proposed Construction	Intel's Proposed Construction
a statement or expression consisting of an operation and its operands (if any)	a programming statement or expression consisting of an operation and its operands (if any), which can be interpreted and executed by a processor in order to perform the specified operation

[3] This term is contained in Claims 1, 6, 7, 12, 17, 19 and 20 of the '072 patent; and Claim 1 of the '244 patent.

The parties largely agree on the construction of "instruction," yet present two issues regarding claim scope: (1) whether the instruction can be interpreted and executed by a processor to perform the operation indicated; and (2) whether an instruction is a programming statement. Stragent contends that Intel's proposed construction imposes limitations that are either redundant to the claim language or unsupported by intrinsic evidence. In particular, Stragent maintains that the claim language already recites that the instruction indicates which polynomial/operation is to be performed; thus, the "interpreted and executed . . . in order to perform the specified operation" language proposed by Intel is redundant. *See* REPLY AT 5-6. Moreover, Stragent notes that the specification does not use the word "programming" to describe "instruction"; while Intel contends that instructions are programming statements—setting forth technical definitions as support—Intel fails to cite to the intrinsic record to show that an instruction is a programming statement. *Id.* at 6. Finally, Stragent maintains that Intel's proposed construction could plausibly confuse the jury, leading jurors to believe that the instructions must perform the specified operation. *Id.*

Intel, however, argues that the intrinsic evidence shows that an instruction is interpreted and executed by a processor. RESPONSE AT 12-13 (citing '072 patent at 3:43-47; 1:66-2:6). Intel further asserts that its proposal does not require that the instructions perform the operations specified, but rather emphasizes that the instructions are "an expression that tells (instructs) the processor which of the operations within the processor's list of possible operations to perform (execute)." *Id.* Finally, Intel points to technical definitions for "instruction," to show that instructions are programming statements interpreted and performed by a processor. *Id.* at 9.

The bulk of Intel's proposal suggests that a processor interprets and executes the instruction.[4] Yet, the claim language already provides context for "instruction." For example, Claim 12 of the '072 patent states that the instructions indicate the operation to be executed: "an instruction *indicating that a CRC operation is to be performed* and indicating which of the first and second circuits is to perform the CRC operation." '072 patent at 7:6-9 (emphasis added). Moreover, Claim 1 states the method comprises "receiving an instruction indicating the CRC operation is to be executed, the instruction including an indication of a polynomial to use in calculating the CRC result." *Id.* at 6:8-10; *see also* '244 patent at 6:8-10. The claim language simply requires that an instruction signal the operation to be performed. Unlike Claim 19, these claims do not recite a processor limitation. *See* ' 072 patent at 8:11-20 ("A network processor comprising . . . means for selecting . . . CRC polynomials to use to perform the CRC operation based on contents of the instruction."). Thus, requiring "instruction" to be interpreted and executed by a processor would render the claim language of Claim 19 superfluous.

In addition, there is no support in the intrinsic record to define "instruction" as a programming statement. Intel does not cite to any portion of the claims or specification to support its position. Rather, Intel cites Stragent's proffered technical definitions and the deposition of the inventor to show that "instructions" are "programming statement[s] or expression[s]." RESPONSE AT 13. Yet, such extrinsic evidence cannot outweigh the lack of any intrinsic evidence on the matter. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1317 (Fed. Cir. 2005) ("[W]hile extrinsic evidence 'can shed useful light on the relevant art,' we have explained that it is 'less significant than the intrinsic record in determining 'the legally operative meaning

[4] The Court notes that it addressed and rejected a similar argument in *Stragent, LLC v. Freescale Semiconductor*, No. 6:10cv224.

of claim language.''" (internal citations omitted)). Thus, the Court declines to construe "instruction" as a programming statement.

Moreover, Intel's proposed language—"can be interpreted and executed by a processor in order to perform the specified operation"—could potentially lead a layperson to believe that the "instruction" performs the "specified operation." Because the parties seem to agree that the instructions (as opposed to, for example, the processor) need not perform the specified operation, any construction for "instruction" shall not imply such a requirement. Therefore, the Court rejects Intel's proposal.

In sum, the claim language simply requires that the instructions indicate the specific operation to be performed; there is no intrinsic support requiring that a processor interpret and execute the instruction. Similarly, an instruction is not limited to a programming statement. Accordingly, the Court finds that an "instruction" is "a statement or expression consisting of an operation and its operands (if any)."

II. "CRC instruction"[5]

Plaintiff's Proposed Construction	Intel's Proposed Construction
a statement or expression consisting of an operation and its operands which indicates that a CRC operation is to be performed	an instruction that, when executed, causes a CRC operation to be performed

The parties dispute whether the "CRC instruction" causes the CRC operations to be performed. *See* PLTFF'S BRIEF AT 15-16; RESPONSE AT 14. Intel contends that when executed, the instruction causes the processor to perform a particular operation, and points to portions of the specification to show that a CRC instruction in particular causes a CRC operation to be performed. RESPONSE AT 14-15 (citing '072 patent at 2:15-20; 4:28-34). Stragent disagrees,

[5] This term is contained in Claim 6 of the '072 patent.

arguing that the Court should adopt its prior provisional construction, which simply states that the instruction *indicates* that a particular operation is to be performed. *See* PLTFF'S BRIEF AT 15.

Intel does not point to particular claim language stating that the CRC instruction causes implementation of the CRC operation. Moreover, the specification does not support a reading that execution of the CRC instruction causes the performance of a CRC operation. Although Intel is correct that the specification discusses the CRC instruction in relation to the CRC operation, the portions cited by Intel are silent regarding causation. Rather, these portions disclose that the CRC instruction *indicates* that a CRC operation is to be performed:

> - Demultiplexer 301 receives the CRC instruction from instruction store 203. *Instructions that indicate that a CRC operation is to be performed* also indicate which of the circuits 305-308 is to perform the CRC operation. Demultiplexer 301 selects the appropriate one of the circuits 305-308 when the instruction indicates a CRC instruction. '072 patent at 4:28-32 (emphasis added);
>
> - The instruction store includes at least one *CRC instruction that indicates that a CRC operation should be performed*. . . . The ALU receives input data for the CRC operation and the CRC instruction, and in response to the CRC instruction, generates the CRC result using the CRC circuit, the input data, and a selected one of the hardwired polynomials, the selected hardwired polynomial being selected based on the CRC instruction. '072 patent at 2:10-20 (emphasis added).

See also id. at 5:39-41 ("The appropriate CRC circuit to use for a particular CRC operation is indicated in the CRC instruction."). Further, although the CRC result is generated in response to the CRC instruction, the CRC instruction does not cause the CRC operation to be performed.

Accordingly, a "CRC instruction" is "a statement or expression consisting of an operation and its operands which indicates that a CRC operation is to be performed."

III. "instruction indicating [that] [the/a] CRC operation is to be [executed/performed/initiated]"[6]

Plaintiff's Proposed Construction	Intel's Proposed Construction
a statement or expression consisting of an operation and its operands which indicates that an operation is to be performed using CRC polynomials to generate a CRC result to be used in error checking	an instruction that, when executed, causes a CRC operation to be [executed/performed/initiated]

Intel maintains that variations of this term also implicate a causation requirement, i.e., that the instruction, when executed, causes a CRC operation to be performed. RESPONSE AT 15. Having resolved the issue above, *see supra* SECTION II, the Court finds that no construction is necessary for "instruction indicating [that] [the/a] CRC operation is to be [executed/performed/initiated]," especially in light of the Court's construction of "instruction," *see supra* SECTION I, and "CRC operation." *See infra* SECTION VIII; *O2 Micro Intern. Ltd. v. Beyond Innovation Tech. Co., Ltd.*, 521 F.3d 1351, 1362 (Fed. Cir. 2008).

IV. "instruction store"[7]

Plaintiff's Proposed Construction	Intel's Proposed Construction
an element that stores instructions in a computing device	a memory that stores a set of program instructions that are fetched for execution

Three issues dominate the dispute between the parties concerning the term "instruction store": (1) whether the instruction store is memory or simply an element; (2) whether the instructions are "program instructions"; and (3) whether said instructions are "fetched for execution." *See* PLTFF'S BRIEF AT 16-17; RESPONSE AT 17-19. Stragent maintains that the instruction store is an element that stores instructions; nothing within the intrinsic evidence suggests that the instruction store is memory from which instructions are fetched. PLTFF'S BRIEF AT 16-17.

[6] This term is contained in Claims 1, 12, 19 and 20 of the '072 patent; and Claim 1 of the '244 patent.
[7] This term is contained in Claims 6, 7 and 20 of the '072 patent.

In contrast, Intel maintains that an instruction store is memory that stores program instructions fetched for execution. RESPONSE AT 17. For support, Intel cites to the specification, references cited on the cover of the ‘072, ‘244 and ‘102 patents, technical definitions, and inventor testimony. *Id.* at 18. Intel contends that “instruction store” is a term of art with specialized meaning, therefore requiring more than the simple definition Stragent offers. *Id.* at 19.

While the construction Stragent offers is simple, it is apt. The claim language and specification describe the instruction store in functional terms, consistently stating that it stores instructions. The specification discloses:

- [A]n instruction store including at least one Cyclic Redundancy Check (CRC) instruction relating to a CRC operation. ‘072 patent at 6:33-35 (Claim 6); *see also* ‘072 patent at 8:21-24 (Claim 20);

- In general, the architecture of network processor 200 is implemented as a set of functional units. . . . More particularly, network processor 200 includes . . . an instruction store 203 . . . . The operation of network processor 200 is controlled by instructions stored in the instruction store 203. ‘072 patent at 3:27-44;

- The instruction store includes at least one CRC instruction that indicates that a CRC operation should be performed. ‘072 patent at 2:10-12;

- The operation of network processor 200 is controlled by instructions stored in instruction store 203. ‘072 patent at 3:42-43.

As the cited portions indicate, an instruction store is a functional unit that stores instructions; there is no intrinsic support for requiring that the instruction store is memory. Moreover, neither the claims, nor the specification impose any further functional limitations, such as requiring that instructions be fetched from the instruction store.[8] Thus, the instruction store is simply as Stragent suggests: an element that stores instructions.

While the specification describes an embodiment where "instruction sequencer 204 controls the transmission of instructions from instruction store 203 to ALU 205," '072 patent at 3:43-45, the specification does not disclose that the instructions are fetched for execution. In other words, there is no indication within the specification that "transmission" equates to "fetching." Moreover, the "fetching" limitation Intel attempts to impose relates to the instructions themselves ("instructions that are fetched for execution"), not the instruction store. Therefore, the "fetching" limitation, even if correct, does not aid in the understanding of "instruction store."

Finally, the Court has addressed Intel's arguments regarding "program instructions." *See supra* SECTION I. Therefore, the Court finds that an "instruction store" is "an element that stores instructions in a computing device."

V. "CRC [output] result"[9]

Plaintiff's Proposed Construction	Intel's Proposed Construction
no construction necessary	the remainder value computed by dividing the input data by a CRC polynomial

At the hearing, the parties agreed that a "CRC result" is "a value equal to the remainder of the input data divided by a CRC polynomial." MARKMAN TRANSCRIPT AT 63:6-12 (Doc. No. 97).

[8] Intel points to references cited on the cover of the patent to show that "instructions are stored in and fetched from memory." RESPONSE AT 18. Intel does not provide any authority explaining why the limitations described within the prior art must be imposed on the claim terms in the '072, '244, and '102 patents.

[9] This term is contained in Claims 1, 2, 4, 6, 8, 12 of the '072 patent; and Claims 1-4 of the '244 patent.

VI. "CRC state data"[10]

Plaintiff's Proposed Construction	Intel's Proposed Construction
information which initializes a CRC circuit prior to performing a CRC operation	information used to initialize the CRC circuit or allow the CRC circuit to perform the CRC operation incrementally

At the claim construction hearing, the parties agreed to the construction of "CRC state data." By agreement, "CRC state data" is "information which initializes a CRC circuit for performing a CRC operation." MARKMAN TRANSCRIPT AT 67:23-68:9.

VII. "crossbar switch"[11]

Plaintiff's Proposed Construction	Intel's Proposed Construction
a device that can simultaneously connect multiple inputs to multiple outputs, wherein each input is connected to each output through a path that contains a single switching node	a device that connects one of multiple inputs to one of multiple outputs

At the hearing, the parties agreed that a "crossbar switch" is "a device that can simultaneously connect one or more of multiple inputs to one or more of multiple outputs." MARKMAN TRANSCRIPT AT 68:12-17.

VIII. "CRC circuit"[12] and "CRC operation"[13]

Term	Plaintiff's Proposed Construction	Intel's Proposed Construction
CRC circuit	a circuit configured to perform error-checking using a CRC polynomial	circuit configured to use a CRC polynomial
CRC operation	an operation performed using CRC polynomials to generate a CRC result to be used in error checking	operation performed using CRC polynomials to generate a CRC result

The parties dispute whether a "CRC circuit" and "CRC operation" are restricted to uses involving error-checking. PLTFF'S BRIEF AT 21; RESPONSE AT 27. Stragent asserts that the

[10] This term is contained in Claim 1 of the '072 patent.
[11] This term is contained in Claims 7, 8 and 10 of the '072 patent.
[12] This term is contained in Claims 1, 6, 12, 16-17, 19 and 21-22 of the '072 patent; and Claim 1 of the '244 patent.
[13] This term is contained in Claims 1, 6, 12, 17, 19 and 22 of the '072 patent; and Claim 1 of the '244 patent.

patents indicate that the purpose of a CRC circuit is to perform a CRC operation to produce a CRC result, which amounts to an error-checking operation. PLTFF'S BRIEF AT 22. Intel contends that CRC circuits and CRC operations are simply mathematical operations that may be used to detect changes in data; however, CRC circuits and operations are not necessarily limited to detecting errors in blocks of data. RESPONSE AT 27. Moreover, Intel notes that some of the independent claims do not refer to CRC circuits performing CRC operations to check for errors. *Id.* at 28 (citing '072 patent at Claims 1, 6 & 12). Rather, Intel contends that because dependent Claim 4 of the '072 patent adds the error-checking limitation, reading such a limitation into the claim terms is unnecessary. *Id.*

The presumption of claim differentiation does not apply under the circumstances, as Intel seems to contend. Claim 4[14] recites multiple limitations, in addition to an error-checking limitation: "The method of claim 3, further comprising: determining that the input data contains errors when the CRC result is a non-zero value." '072 patent at 6:26-28. As the cited portion illustrates, Claim 4 recites an error-checking limitation, as well as an additional limitation regarding "a non-zero value." Therefore, Claim 4 adds more than one significant limitation to the independent claim. Accordingly, the Court cannot conclude that the presence of an error-checking limitation in Claim 4 gives rise to the presumption that such a limitation is not present in Claim 1. *See SunRace Roots Enter. Co., Ltd. v. SRAM Corp.*, 336 F.3d 1298, 1303 (Fed. Cir. 2003) ("[The] presumption [of claim differentiation] is especially strong when the limitation in dispute is the only meaningful difference between an independent and dependent claim, and one party is urging that the limitation in the dependent claim should be read into the independent claim.").

[14] Claim 4 depends on Claim 3, which depends on independent Claim 1.

While Intel is correct that some of the claims do not explicitly recite that a CRC circuit and CRC operations are used to check for errors, *compare* '072 patent at Claim 1 *with* '244 patent at Claim 6, these claims do not indicate that that the CRC operations are performed for tasks other than error-checking. Moreover, the specification indicates that the patentee intended for the patented technology to be a more efficient way to perform CRC calculations for error-checking:

> One operation that may be implemented as a special instruction in a specialized network processor is a *Cyclic Redundancy Check (CRC) error checking operation. CRC is the most common method of error detection for many data communication protocols*. A CRC value is computed for a packet and attached to the packet during transmission. The device receiving the packet can *verify the integrity of the packet by re-calculating the packet's CRC value and comparing it to the attached value*.
>
> Given the wide spread use of *CRC based error checking techniques* in data communication networks, there is a need in the art for special purpose processor that can perform CRC calculations in an efficient manner.

'072 patent at 1:45-58 (emphasis added); *see also* '072 patent at 3:58-59 ("In general, CRC operations are used as a way of detecting small changes, such as transmission errors, in blocks of data."); 3:66-4:2 ("When the data is received (or recovered from storage) the CRC operation can be reapplied, and the latest result compared to the original result. If no error has occurred, the CRC results should not match."); 5:31-34 ("When using a CRC circuit to subsequently check the integrity of the data, the data is concatenated with the CRC result value and input to the CRC circuit 400. If there are no errors in the data value, the new CRC result should be zero."). As the cited portions indicate, the patentee framed the invention in terms of a more efficient way of using CRC calculations to check for errors during data transmission.

Intel suggests that CRC calculations are used to detect small changes in data and may be used in hashing methods that are not useful to check for errors. However, Intel does not explain how detecting changes in data differs from error-checking as disclosed in the patents. In fact, Intel acknowledges that in general, the changes detected by CRC operations will be errors. RESPONSE AT 27. As for hashing techniques, while it may be that, in isolation, CRC operations are used to hash values, Intel does not explain why the CRC operations, as disclosed in this intrinsic record, relate to hashing methods.

Although CRC calculations may be used for other purposes, the specification contemplates the use of CRC calculations for error-checking purposes. The parties generally agree that a CRC operation is an operation performed using a CRC polynomial to generate a CRC result. Moreover, the patent describes using CRC circuits hardwired with polynomials to execute a CRC operation to produce a CRC result. '072 patent at 4:11-28. The specification further extolls the error-checking advantages of using longer CRC polynomials to produce a CRC result:

> Different CRC polynomials are possible in different CRC implementations. The CRC polynomials are generally designed and constructed to have desirable *error-detection properties*. In general, longer polynomials provide more assurance of data accuracy and are fully useable over larger amounts of data; however, long polynomials also produce longer remainder values, which add additional *error-checking* overhead to the data.

'072 patent at 4:3-10 (emphasis added). Thus, using a CRC polynomial to generate a CRC result—which the parties generally agree is a CRC operation—implies an error detection process. Similarly, the parties largely agree that a CRC circuit is configured to use a CRC polynomial. As shown above, the specification touts the advantages of CRC polynomials in checking for errors to assure data accuracy. No disclosure of using these polynomials for any other purpose can be found in the intrinsic record. Therefore, it follows that a CRC circuit and CRC operations, which both use CRC polynomials, are used to check for errors.

Accordingly, the Court finds that a "CRC circuit" is "a circuit configured to perform error-checking using a CRC polynomial." Moreover, a "CRC operation" is "an operation performed using CRC polynomials to generate a CRC result to be used in error checking."

CONCLUSION

For the foregoing reasons, the court adopts the constructions set forth above.

So ORDERED and SIGNED this 8th day of August, 2013.

John D Love

JOHN D. LOVE
UNITED STATES MAGISTRATE JUDGE